Design Optimization

Different techniques to make
designs better!

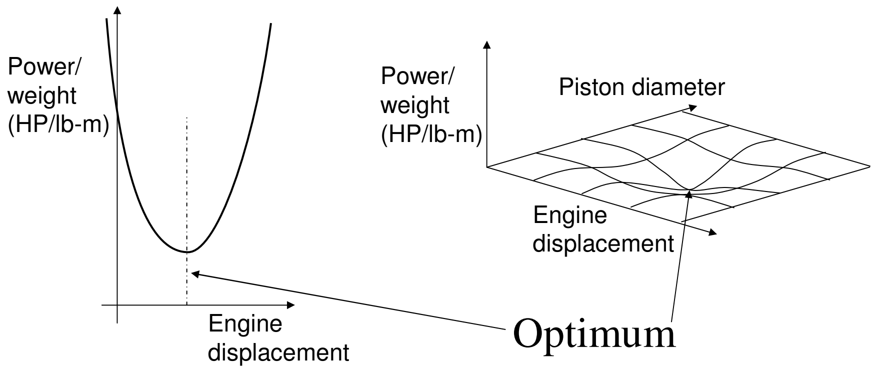
What is an Optimization Problem?

*"**Optimization** is a process of selecting or converging onto a final solution amongst a number of possible options, such that a certain requirement or a set of requirements is **best** satisfied."*

I.e., you want a design in which some quantifiable property is **minimized** or **maximized** (e.g., strength, weight, strength-to-weight ratio)

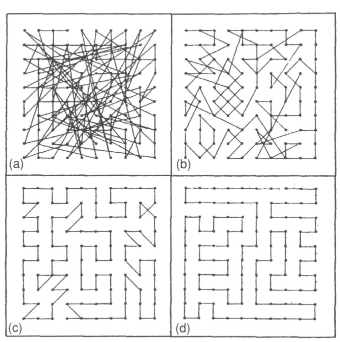
Some Types of Optimization Problems

- Continuous real parameter(s):

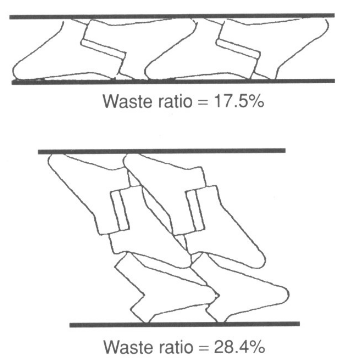


Some Types of Optimization Problems

- Combinatorial:



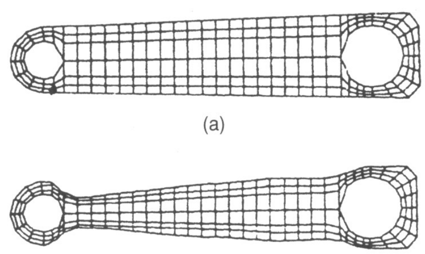
Geometric:



Figures from: K. Lee, "Principles of CAD/CAM/CAE Systems," Addison-Wesley, 1999

Some Types of Optimization Problems

• Structural



(a)

(b)

E.g., Minimize weight, maximize strength

Figure from: K. Lee, "Principles of CAD/CAM/CAE Systems," Addison-Wesley, 1999

Formulating the Problem

To have a set of possible solutions, the design must be parameterized. The objective function must be defined in terms of those parameters.

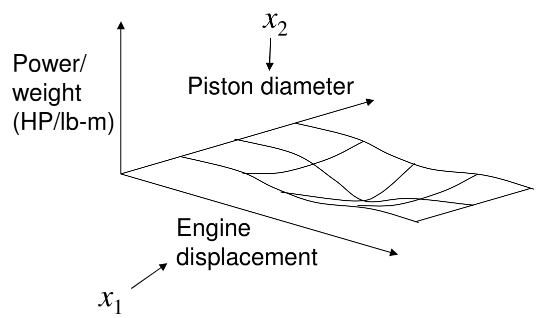
Formulation steps:

a) Identify design variables
   (e.g., *engine_displacement*, *piston_diameter*)
b) Define objective function
   (e.g., maximize *power_to_weight_ratio*)
c) Identify constraints
   (e.g., 1 inch ≤ *piston_diameter* ≤ 12 inch)

Formulating the Problem

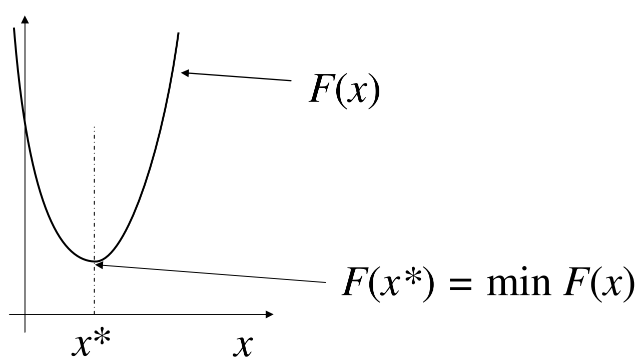
$F(x_1, x_2)$

$F(\mathbf{X})$



$$\mathbf{X} = \begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix}$$

Formulating the Problem



Formulating the Problem

Mathematically, you need to select:

a) Design Variables (vector $\mathbf{X}$, solution $\mathbf{X}^* \in R^n$)

$$\text{e.g., } \mathbf{X} = \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix}, \mathbf{X}^* = \begin{Bmatrix} 5.4 \\ 6.0 \\ 9.8 \end{Bmatrix}$$

This means $\mathbf{X}^*$ is a vector of n real numbers.

b) Objective Function ($F(\mathbf{X})$)

$$\mathbf{X}^* \in R^n \text{ so that } F(\mathbf{X}^*) = \min F(\mathbf{X})$$

In other words, the solution is the vector of real numbers $\mathbf{X}^*$ for which $F(\mathbf{X})$ is minimum.

Formulating the Problem

c) Constraints

- bounds $\mathbf{X}_l \le \mathbf{X}^* \le \mathbf{X}_u$ e.g., $\begin{Bmatrix} 5 \\ 5 \\ 5 \end{Bmatrix} \le \begin{Bmatrix} x_1^* \\ x_2^* \\ x_3^* \end{Bmatrix} \le \begin{Bmatrix} 10 \\ 10 \\ 10 \end{Bmatrix}$

- inequality $G_i(\mathbf{X}^*) \ge 0 \qquad i = 1, 2, ..., m$

$$(\text{e.g., } x_1^* + x_2^* \ge 0 \;\; \therefore \;\; G_1(\mathbf{X}^*) = x_1^* + x_2^*)$$

- equality $H_j(\mathbf{X}^*) = 0 \qquad j = 1, 2, ..., q$

$$(\text{e.g., } x_2^* - x_3^* = 0 \;\; \therefore \;\; H_1(\mathbf{X}^*) = x_2^* - x_3^*)$$

Formulating the Problem

Constraints act as a guide for the optimization problem.

Bounds: Are direct limits on the values a parameter can take (e.g., $5 \leq x_1 \leq 10$.)

Inequality: Are expressions that limit the values parameters can take (e.g., $x_1 - x_2 - 5 \geq 0$.)

Equality: These reduce one design variable for each equality constraint. (e.g., $x_1 - x_3 - 5 = 0$.)

Structural Optimization

Structural optimization is a specific class of optimization problems that uses an FE analysis as part of the **objective function or constraints**.

Structural optimization involves three elements:

1. Automatic modification of structure/FE mesh
2. FE analysis
3. Optimization algorithm

Structural Optimization

Three ways of modifying structures:

– Parameter optimization
  - Concept: change solid model parameters.
  - Procedure:
    1. Create initial solid model.
    2. Create initial FE model from solid model.
    3. Execute FEA.
    4. Evaluate objective function and constraints.
    5. Stop if design is optimal; otherwise:
       a. Change dimension or parameter in solid model.
       b. Re-execute solid construction.
       c. Re-mesh FE model.
       d. Return to Step 3.

Structural Optimization

– Shape Optimization
  - Concept: Move part boundaries.
  - Procedure (automated):
    1. Create initial FE model; or create surface/solid model and mesh
    2. Execute FEA.
    3. Evaluate objective function and constraints.
    4. Stop if design is optimal; otherwise:
       a. Move node in FE model; or move control point on surface and remesh
       b. Return to Step 2.

– Topology Optimization
  - Concept: change density of material regions to form shape and topology.
  - Procedure:
    1. Initialize densities in FE model.
    2. Execute FEA
    3. Evaluate objective function and constraints.
    4. Stop if design is optimal; otherwise:
       a. Correct densities
       b. Return to Step 2.

Shape Optimization

Optimal truss design: Node locations and cross-section properties are the design variables.

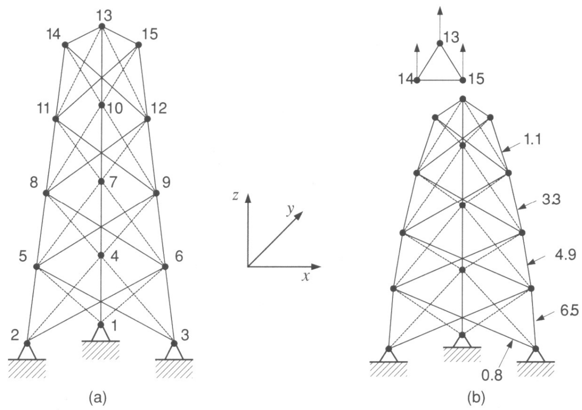


Figure from: K. Lee, "Principles of CAD/CAM/CAE Systems," Addison-Wesley, 1999

Shape Optimization

The locations of the FE nodes or the B-Spline control points are the design variables.

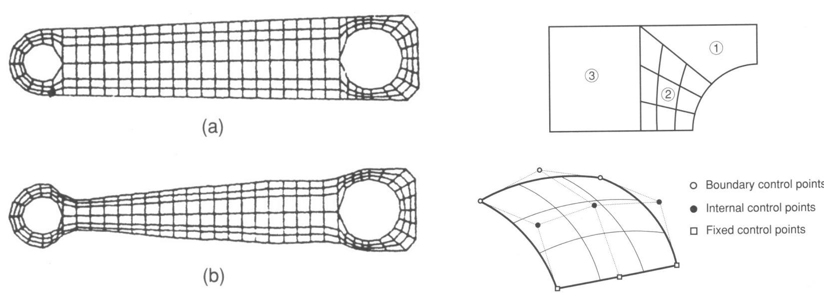


Figure from: K. Lee, "Principles of CAD/CAM/CAE Systems," Addison-Wesley, 1999

Topology Optimization

In topology optim-
ization, the design variables are the amounts of material in each cell.

Material is only added where it is needed to carry loads.

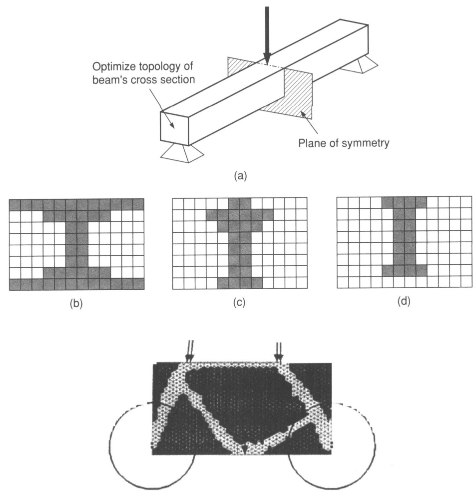


Figure from: K. Lee, "Principles of CAD/CAM/CAE Systems," Addison-Wesley, 1999

Topology Optimization

Besides allowing for size and shape changes, topology optim-
ization allows voids to appear or disappear.

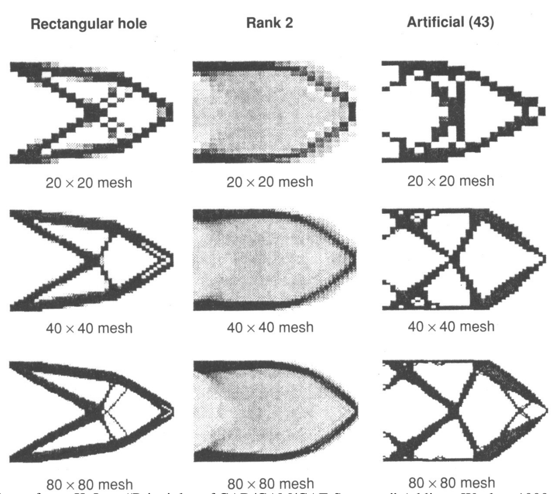


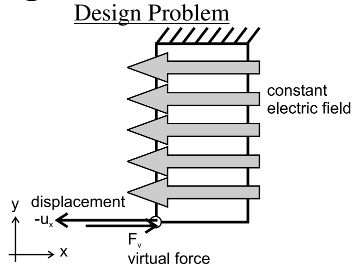
Figure from: K. Lee, "Principles of CAD/CAM/CAE Systems," Addison-Wesley, 1999

Topological Optimization

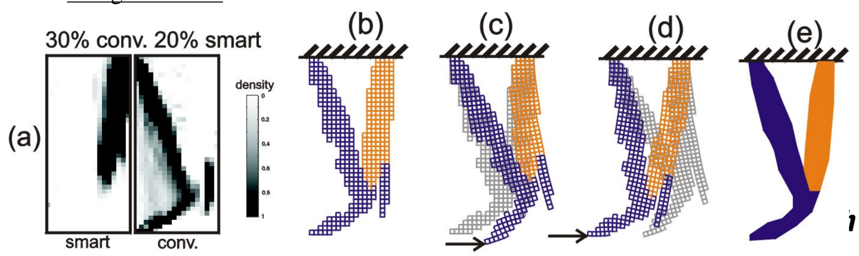
Optimal Smart Structure Design at MTU!

Simultaneous optimization of:
- conventional material structure,
- embedded sensors,
- embedded actuators, and
- controller design.

Design Problem



Design Solution



Choosing a Solution Method

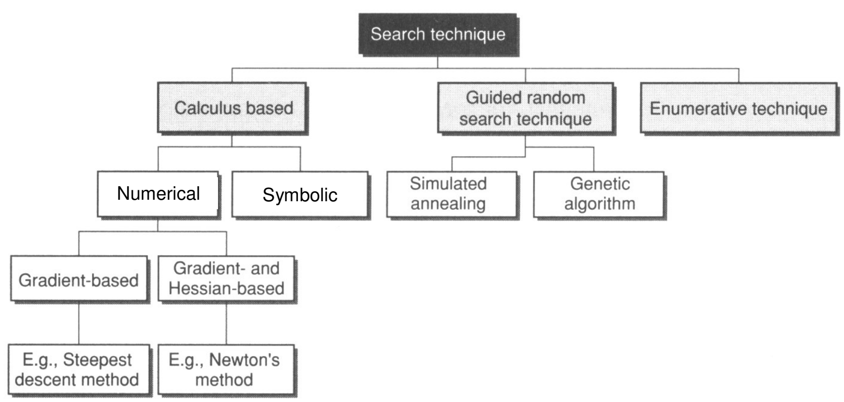


Figure from: K. Lee, "Principles of CAD/CAM/CAE Systems," Addison-Wesley, 1999

Symbolic Solution

- The symbolic solution is found by solving

$$\frac{\partial F}{\partial x_i} = 0 \quad \text{such that} \left(\frac{\partial^2 F}{\partial x_1 \partial x_2} \right)^2 - \frac{\partial^2 F}{\partial x_1^2} \frac{\partial^2 F}{\partial x_2^2} < 0 \quad (\text{for } n = 2)$$

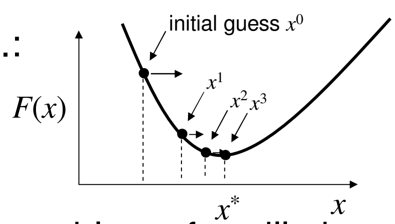
- E.g., $F(\mathbf{X}) = x_1^2 + (x_2 - 5)^2$ $\qquad x_1 = 0, \; x_2 = 5$

$$\frac{\partial F}{\partial x_1} = 2x_1 = 0, \quad \frac{\partial F}{\partial x_2} = 2(x_2 - 5) = 0 \qquad (0)^2 - 2 \cdot 2 < 0$$

- However, most engineering optimization problems have objective functions that are too complicated, with many constraints, and with too many design variables to make this method tractable.

Gradient-Based Solving

- Gradient-Based methods are iterative. They choose a better solution by following the downward slope of the curve/surface given by: $\frac{\partial F}{\partial x_i}$
- E.g.:



- The problem of oscillations in the steepest descent method is solved by Newton's method.

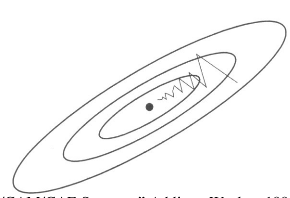
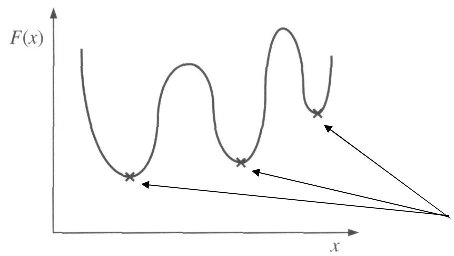


Figure from: K. Lee, "Principles of CAD/CAM/CAE Systems," Addison-Wesley, 1999

Local Minima

- Gradient-based methods do not work well when there are several local minima:



- The **Simulated Annealing** and **Genetic Algorithm** methods were introduced to solve this problem.

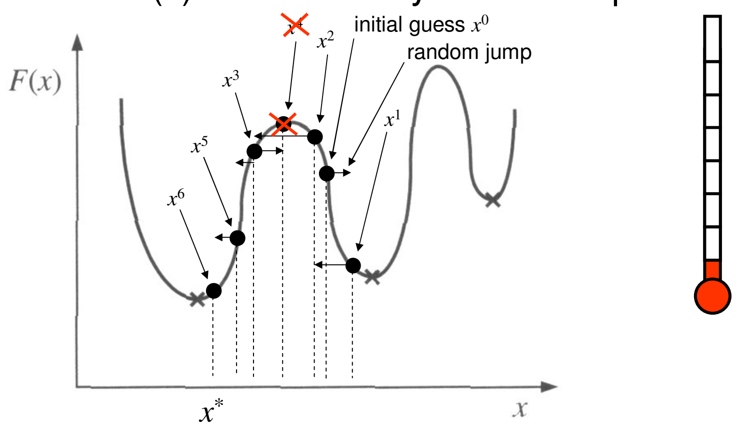
Figure from: K. Lee, "Principles of CAD/CAM/CAE Systems," Addison-Wesley, 1999

Simulated Annealing

- Concept: when temperature is high, solution will jump around a lot but as the temperature cools the solution will settle into a local minimum
- Procedure:
  1. Set initial temperature and initial guess at solution.
  2. Evaluate objective and constraints.
  3. Stop if minimum temperature reached; otherwise:
     a. Take a random jump to a neighboring solution.
     b. Evaluate objective and constraints.
     c. If solution is better then accept it; otherwise:
        a. Generate a random number
        b. If the random number is less than the temperature then accept the new solution
  4. Decrease the temperature
  5. Go to step 3.

Simulated Annealing

- As the "temperature" cools, jumps to higher values of F(x) are less likely to be accepted.



Genetic Algorithm

- Concept: have a large set of solutions; combine the best solutions to get even better solutions
- Procedure:
  1. Create a population of initial random solutions.
  2. Evaluate objective and constraints for each.
  3. Create new solutions for next generation by:
     a. Randomly crossing two solutions
     b. Randomly creating mutated solutions
  4. Evaluate objective and constraints for each new solution
  5. Get rid of the worst solutions so that the total number of solutions stays the same.
  6. Increment the generation number.
  7. Stop if the maximum number of generations is reached. Otherwise go to step 3.

Genetic Algorithm

- In genetic algorithms, the parameters in ***X*** are represented in a binary form:

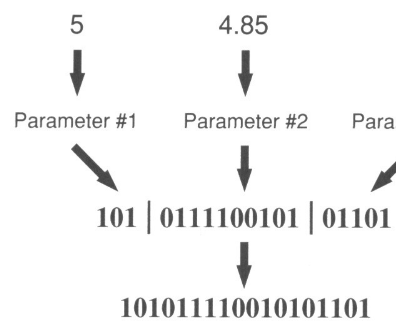


Figure from: K. Lee, "Principles of CAD/CAM/CAE Systems," Addison-Wesley, 1999

Genetic Algorithm

New solutions are added by modifying existing solutions randomly using:

1. Cross-over:

New children with gene segments from both parents

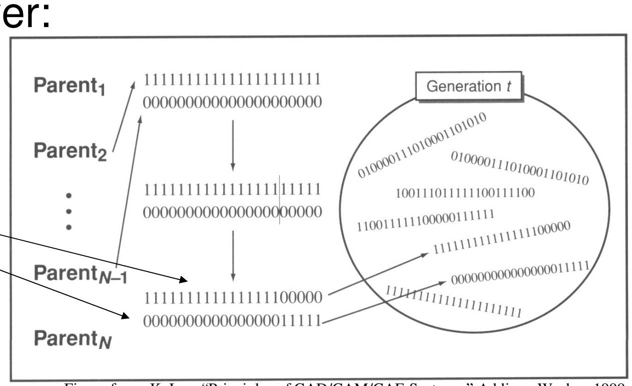
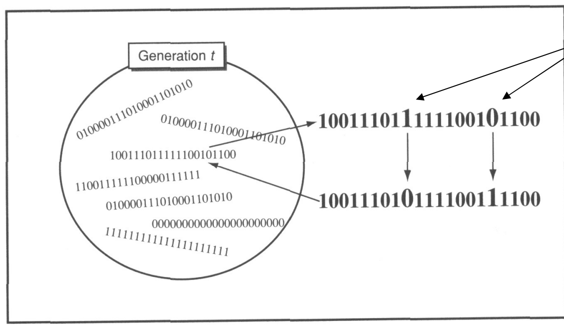


Figure from: K. Lee, "Principles of CAD/CAM/CAE Systems," Addison-Wesley, 1999

Genetic Algorithm

2. Mutation:



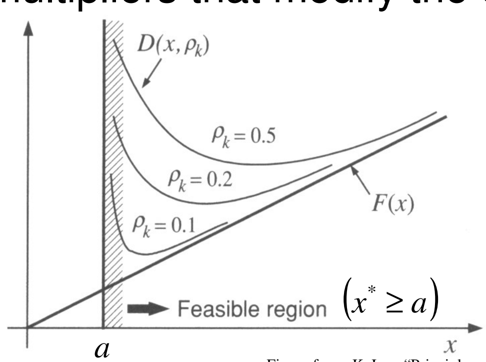
These bits chosen randomly to be changed.

After each iteration, only the solutions that best meet the objective function are kept.

Figure from: K. Lee, "Principles of CAD/CAM/CAE Systems," Addison-Wesley, 1999

Constraint Handling

A common way of handling constraints is to introduce penalty parameters (e.g. ρ_k) as multipliers that modify the objective function.



(Optimize $D(x, \rho_k)$ instead of $F(x)$.)

Figure from: K. Lee, "Principles of CAD/CAM/CAE Systems," Addison-Wesley, 1999